AMENDMENTS TO THE CLAIMS

1. (currently amended) A method of software change modeling for nodes in a distributed network of nodes, the method comprising the computer-implemented steps of:
providing a master node;
receiving a software update for a node on said master node;
wherein the software update contains a software package or a set of software packages;
wherein a software package contains at least one software module with corresponding software dependency information;
wherein said master node notifies said node that a software update is being requested; [[and]]
wherein said master node passes said node identities of software package(s) to be updated and software dependency information; and
wherein said node determines, using the software dependency information, running processes on said node that will be affected by the software update.

2. (canceled).

3. (currently amended) A method as recited in Claim [[2]] 1, wherein said node notifies processes that have indicated interest in software updates that the software update is being requested; wherein each notified process evaluates the effect that the software update will have on its operation; wherein if any of the processes determine that the software update will degrade or have a negative impact on said node's normal operation, the process returns a veto to said node along with reasons why; and wherein if a process finds that the software update will have no negative effects, the process returns an acceptance of the software update to said node.

4. (original) A method as recited in Claim 3, wherein said node waits for all of the notified processes to return the results of their evaluations and once all of the processes have reported to said node, said node notifies said master node if any of the processes have vetoed the software update along with their reasons.

5. (original) A method as recited in Claim 4, wherein said master node displays node identifiers and the processes that have vetoed the software update along with their reasons to a user.

6. (original) A method as recited in Claim 1, wherein a user initiates a software update by installing an image containing the software update onto said master node.

7. (original) A method as recited in Claim 6, wherein the user indicates what nodes and which software package(s) are to be updated.

8. (original) A method as recited in Claim 1, wherein a software package indicates the type of node to which it applies.

9. (original) A method as recited in Claim 1, wherein the software update contains a list of software packages destined for each node.

10. (original) A method as recited in Claim 1, wherein a software package contains version information, dependency information, and other metadata information pertaining to software in the package.

11. (original) A method as recited in Claim 10, wherein the metadata includes a list of application program interface (API) providers and consumers.

12. (currently amended) An apparatus of software change modeling for nodes in a distributed network of nodes, comprising:
a master node;
means for receiving a software update for a node on said master node;
wherein the software update contains a software package or a set of software packages;
wherein a software package contains at least one software module with corresponding software dependency information;

wherein said master node notifies said node that a software update is being requested; [[and]]

wherein said master node passes said node identities of software package(s) to be updated and software dependency information; and

wherein said node determines, using the software dependency information, running processes on said node that will be affected by the software update.

13. (canceled).

14. (currently amended) An apparatus as recited in Claim [[13]] 12, wherein said node notifies processes that have indicated interest in software updates that the software update is being requested; wherein each notified process evaluates the effect that the software update will have on its operation; wherein if any of the processes determine that the software update will degrade or have a negative impact on said node's normal operation, the process returns a veto to said node along with reasons why; and wherein if a process finds that the software update will have no negative effects, the process returns an acceptance of the software update to said node.

15. (original) An apparatus as recited in Claim 14, wherein said node waits for all of the notified processes to return the results of their evaluations and once all of the processes have reported to said node, said node notifies said master node if any of the processes have vetoed the software update along with their reasons.

16. (original) An apparatus as recited in Claim 15, wherein said master node displays node identifiers and the processes that have vetoed the software update along with their reasons to a user.

17. (original) An apparatus as recited in Claim 12, wherein a user initiates a software update by installing an image containing the software update onto said master node.

18. (original) An apparatus as recited in Claim 17, wherein the user indicates what nodes and which software package(s) are to be updated.

19. (original) An apparatus as recited in Claim 12, wherein a software package indicates the type of node to which it applies.

20. (original) An apparatus as recited in Claim 12, wherein the software update contains a list of software packages destined for each node.

21. (original) An apparatus as recited in Claim 12, wherein a software package contains version information, dependency information, and other metadata information pertaining to software in the package.

22. (original) An apparatus as recited in Claim 21, wherein the metadata includes a list of application program interface (API) providers and consumers.

23. (currently amended) A computer-readable storage medium carrying one or more sequences of instructions for software change modeling for nodes in a distributed network of nodes, which instructions, when executed by one or more processors, cause the one or more processors to carry out the steps of:
providing a master node;
receiving a software update for a node on said master node;
wherein the software update contains a software package or a set of software packages;
wherein a software package contains at least one software module with corresponding software dependency information;
wherein said master node notifies said node that a software update is being requested; [[and]]
wherein said master node passes said node identities of software package(s) to be updated and software dependency information; and

wherein said node determines running processes on said node that will be affected by the software update using the software dependency information.

24. (canceled).

25. (currently amended) A computer-readable storage medium as recited in Claim [[24] 23, wherein said node notifies processes that have indicated interest in software updates that the software update is being requested; wherein each notified process evaluates the effect that the software update will have on its operation; wherein if any of the processes determine that the software update will degrade or have a negative impact on said node's normal operation, the process returns a veto to said node along with reasons why; and wherein if a process finds that the software update will have no negative effects, the process returns an acceptance of the software update to said node.

26. (currently amended) A computer-readable storage medium as recited in Claim 25, wherein said node waits for all of the notified processes to return the results of their evaluations and once all of the processes have reported to said node, said node notifies said master node if any of the processes have vetoed the software update along with their reasons.

27. (currently amended) A computer-readable storage medium as recited in Claim 26, wherein said master node displays node identifiers and the processes that have vetoed the software update along with their reasons to a user.

28. (currently amended) A computer-readable storage medium as recited in Claim 23, wherein a user initiates a software update by installing an image containing the software update onto said master node.

29. (currently amended) A computer-readable storage medium as recited in Claim 28, wherein the user indicates what nodes and which software package(s) are to be updated.

30. (currently amended) A computer-readable storage medium as recited in Claim 23, wherein a software package indicates the type of node to which it applies.

31. (currently amended) A computer-readable storage medium as recited in Claim 23, wherein the software update contains a list of software packages destined for each node.

32. (currently amended) A computer-readable storage medium as recited in Claim 23, wherein a software package contains version information, dependency information, and other metadata information pertaining to software in the package.

33. (currently amended) A computer-readable storage medium as recited in Claim 32, wherein the metadata includes a list of application program interface (API) providers and consumers.

34. (currently amended) A method of software change modeling of networked nodes on a computer system, the method comprising the computer-implemented steps of:
providing a software update simulator on said computer system;
simulating processes from at least one node on said computer system;
wherein each functional process ~~from said node~~ that is simulated is a minimal version of a functional process that runs on said node; and
receiving a software update for [[a]] said node by said software update simulator;
wherein the software update contains a software package or a set of software packages;
wherein a software package contains at least one software module with corresponding software dependency information;

wherein said software update simulator notifies a control process for said node that a software update is being requested; and

wherein said software update simulator passes said control process identities of software package(s) to be updated and software dependency information.

35. (original) A method as recited in Claim 34, wherein said control process determines running functional node processes that will be affected by the software update using the software dependency information.

36. (original) A method as recited in Claim 35, wherein said control process notifies processes that have indicated interest in software updates that the software update is being requested; wherein each notified process evaluates the effect that the software update will have on its operation; wherein if any of the processes determine that the software update will degrade or have a negative impact on said node's normal operation, the process returns a veto to said control process along with reasons why; and wherein if a process finds that the software update will have no negative effects, the process returns an acceptance of the software update to said control process.

37. (original) A method as recited in Claim 36, wherein said control process waits for all of the notified processes to return the results of their evaluations and once all of the processes have reported to said control process, said control process notifies said software update simulator if any of the processes have vetoed the software update along with their reasons.

38. (original) A method as recited in Claim 37, wherein said software update simulator displays node identifiers and the processes that have vetoed the software update along with their reasons to the user.

39. (original) A method as recited in Claim 34, wherein a user initiates a software update by loading an image containing the software update into said software update simulator.

40. (original) A method as recited in Claim 39, wherein the user indicates what nodes and which software package(s) are to be updated.

41. (original) A method as recited in Claim 34, wherein a software package contains version information, dependency information, and other metadata information pertaining to software in the package.

42. (original) A method as recited in Claim 41, wherein the metadata includes a list of application program interface (API) providers and consumers.

43. (original) A method of software change modeling of nodes in a network of nodes on a computer system, the method comprising the computer-implemented steps of:
providing a software update simulator on said computer system;
wherein said software simulator runs software components normally run on a master node in the network of nodes;
wherein a user loads a node's current software configuration into said software simulator by loading current software modules installed on a node;
wherein the user requests a simulation of a software update by loading an updated software image into said simulator;
wherein the software image contains a software package or a set of software packages;
wherein a software package contains at least one software module with corresponding software dependency information;
wherein said software simulator calculates the software update's impact on said node using the current software configuration of said node; and
displaying the calculation's results to the user.

44. (original) A method as recited in Claim 43, wherein the user also indicates to said software simulator the type of node being analyzed.

45. (original) A method as recited in Claim 43, wherein said software update is a software downgrade where modules are being removed.

46. (original) An apparatus of software change modeling of nodes in a network of nodes on a computer system, comprising:
a software update simulator on said computer system;
wherein said software simulator runs software components normally run on a master node in the network of nodes;
wherein a user loads a node's current software configuration into said software simulator by loading current software modules installed on a node;
wherein the user requests a simulation of a software update by loading an updated software image into said simulator; and
wherein the software image contains a software package or a set of software packages;
wherein a software package contains at least one software module with corresponding software dependency information;
wherein said software simulator calculates the software update's impact on said node using the current software configuration of said node; and
means for displaying the calculation's results to the user.

47. (original) An apparatus as recited in Claim 46, wherein the user also indicates to said software simulator the type of node being analyzed.

48. (original) An apparatus as recited in Claim 46, wherein said software update is a software downgrade where modules are being removed.

49. (currently amended) A computer-readable storage medium carrying one or more sequences of instructions for software change modeling of nodes in a network of nodes on a computer system, which instructions, when executed by one or more processors, cause the one or more processors to carry out the steps of:
providing a software update simulator on said computer system;

wherein said software simulator runs software components normally run on a master node in the network of nodes;

wherein a user loads a node's current software configuration into said software simulator by loading current software modules installed on a node;

wherein the user requests a simulation of a software update by loading an updated software image into said simulator;

wherein the software image contains a software package or a set of software packages;

wherein a software package contains at least one software module with corresponding software dependency information;

wherein said software simulator calculates the software update's impact on said node using the current software configuration of said node; and

displaying the calculation's results to the user.

50. (currently amended) A computer-readable storage medium as recited in Claim 49, wherein the user also indicates to said software simulator the type of node being analyzed.

51. (currently amended) A computer-readable storage medium as recited in Claim 49, wherein said software update is a software downgrade where modules are being removed.

52. (new) An apparatus for software change modeling of networked nodes on a computer system, the apparatus comprising:

a software update simulator on said computer system;

means for simulating processes from at least one node on said computer system;

wherein each functional process that is simulated is a minimal version of a functional process that runs on said node; and

means for receiving a software update for said node by said software update simulator;

wherein the software update contains a software package or a set of software packages;

wherein a software package contains at least one software module with corresponding software dependency information;

wherein said software update simulator notifies a control process for said node that a software update is being requested; and

wherein said software update simulator passes said control process identities of software package(s) to be updated and software dependency information.

53. (new) A computer-readable storage medium carrying one or more sequences of instructions for software change modeling of networked nodes on a computer system, which instructions, when executed by one or more processors, cause the one or more processors to perform:

providing a software update simulator on said computer system;

simulating processes from at least one node on said computer system;

wherein each functional process that is simulated is a minimal version of a functional process that runs on said node; and

receiving a software update for said node by said software update simulator;

wherein the software update contains a software package or a set of software packages;

wherein a software package contains at least one software module with corresponding software dependency information;

wherein said software update simulator notifies a control process for said node that a software update is being requested; and

wherein said software update simulator passes said control process identities of software package(s) to be updated and software dependency information.

54. (new) An apparatus comprising:

a master node;

one or more processors;

one or more sequences of instructions which, when executed by the one or more processors, causes the one or more processors to perform receiving a software update for a node on said master node;

wherein the software update contains a software package or a set of software packages;

wherein a software package contains at least one software module with corresponding software dependency information;

wherein said master node notifies said node that a software update is being requested;

wherein said master node passes said node identities of software package(s) to be updated and software dependency information; and

wherein said node determines, using the software dependency information, running processes on said node that will be affected by the software update.

55. (new) An apparatus as recited in Claim 54, wherein said node notifies processes that have indicated interest in software updates that the software update is being requested; wherein each notified process evaluates the effect that the software update will have on its operation; wherein if any of the processes determine that the software update will degrade or have a negative impact on said node's normal operation, the process returns a veto to said node along with reasons why; and wherein if a process finds that the software update will have no negative effects, the process returns an acceptance of the software update to said node.

56. (new) An apparatus as recited in Claim 55, wherein said node waits for all of the notified processes to return the results of their evaluations and once all of the processes have reported to said node, said node notifies said master node if any of the processes have vetoed the software update along with their reasons.

57. (new) An apparatus as recited in Claim 56, wherein said master node displays node identifiers and the processes that have vetoed the software update along with their reasons to a user.

58. (new) An apparatus comprising:

a software update simulator on a computer system;

one or more processors;

one or more sequences of instructions which, when executed by the one or more processors, cause the one or more processors to perform:

simulating processes from at least one node on said computer system, wherein each functional process that is simulated is a minimal version of a functional process that runs on said node; and

receiving a software update for said node by said software update simulator;

wherein the software update contains a software package or a set of software packages;

wherein a software package contains at least one software module with corresponding software dependency information;

wherein said software update simulator notifies a control process for said node that a software update is being requested; and

wherein said software update simulator passes said control process identities of software package(s) to be updated and software dependency information.

59. (new) An apparatus as recited in Claim 58, wherein said control process determines running functional node processes that will be affected by the software update using the software dependency information.

60. (new) An apparatus as recited in Claim 59, wherein said control process notifies processes that have indicated interest in software updates that the software update is being requested; wherein each notified process evaluates the effect that the software update will have on its operation; wherein if any of the processes determine that the software update will degrade or have a negative impact on said node's normal operation, the process returns a veto to said control process along with reasons why; and wherein if a process finds that the software update will have no negative effects, the process returns an acceptance of the software update to said control process.

61. (new) An apparatus as recited in Claim 60, wherein said control process waits for all of the notified processes to return the results of their evaluations and once all of the processes have reported to said control process, said control process notifies said software update simulator if any of the processes have vetoed the software update along with their reasons.

62. (new) An apparatus as recited in Claim 61, wherein said software update simulator displays node identifiers and the processes that have vetoed the software update along with their reasons to the user.

63. (new) An apparatus comprising:

a software update simulator on a computer system;

wherein said software simulator runs software components normally run on a master node in the network of nodes;

wherein a user loads a node's current software configuration into said software simulator by loading current software modules installed on a node;

wherein the user requests a simulation of a software update by loading an updated software image into said simulator;

wherein the software image contains a software package or a set of software packages;

wherein a software package contains at least one software module with corresponding software dependency information;

wherein said software simulator calculates the software update's impact on said node using the current software configuration of said node;

one or more processors; and

one or more sequences of instructions which, when executed by the one or more processors, cause the one or more processors to perform displaying the calculation's results to the user.

64. (new) An apparatus as recited in Claim 63, wherein said software update is a software downgrade where modules are being removed.